Scalable Anonymous Group Communication in the Anytrust Model

David Isaac Wolinsky, Henry Corrigan-Gibbs, and Bryan Ford
Yale University
{david.wolinsky, henry.corrigan-gibbs, bryan.ford}@yale.edu

Aaron Johnson
U.S. Naval Research Laboratory
aaron.m.johnson@nrl.navy.mil

ABSTRACT

Anonymous communication capabilities are useful and desirable, but practical onion routing approaches are vulnerable to traffic analysis and DoS attacks—especially against a powerful adversary, such as a repressive government or compromised ISP. To fill this gap we introduce D3, the first practical anonymous group communication system offering anonymity and disruption resistance against strong traffic analysis and collusion attacks, with scalability to hundreds of group members and quick response to churn. D3 builds on a trust model we call *anytrust*. Anytrust is a decentralized client/server network model, in which each of many clients—representing group members—trust only that *at least one* of a smaller but diverse set of "servers" or "super-peers" behaves honestly, but *clients need not know which server to trust.* By combining and adapting verifiable shuffle and DC-nets techniques to anytrust, D3 achieves short shuffle latencies and efficient tree-based DC-nets ciphertext combining, while guaranteeing message anonymity and integrity, transmission proportionality among group members, and adaptability to both network/node failures and active disruption. Experiments with a working prototype demonstrate that D3 is practical at scales infeasible in prior systems offering comparable security.

1. INTRODUCTION

As use of the Internet expands, individual privacy risks continue to increase, often leading to embarrassment, identity theft [7], and threats to freedom of speech [4]. The right and ability to maintain anonymity online is widely valued as a means to bolster democratic society [26]. In particular, anonymity enables individuals to exercise freedoms of speech and association without fear of reprisal [33]. Mix-nets [10], onion routing [16], and file sharing protocols [12, 5] address these goals, but are difficult to protect against traffic analysis [29] and active DoS attacks [8], especially when the adversary may be powerful, such as a compromised ISP or repressive government [13]. "Dining cryptographers" or DC-nets [11, 31, 14] resist traffic analysis but present unsolved scalability challenges, especially in the presence of network churn. Voting schemes [28, 2], typically designed to shuffle or count small multiple-choice ballots, are insufficient for general anonymous communication.

To enable users to participate in online forums while protecting their identities from powerful adversaries, we introduce D3, the first practical, general-purpose anonymous group communication system offering provable security against traffic analysis, scalability to groups containing hundreds of members or more, and prompt recovery from both network churn and active disruption attacks. The key technical idea that enables D3 to achieve traffic analysis resistance and scalability in combination is the novel use of a client/server network model we call *anytrust*. A D3 group consists of a potentially large set of *client* nodes representing group members (users), and a typically smaller set of *server* nodes that coordinate and facilitate anonymous communication. We use the terms "client" and "server" here merely to express protocol roles. Volunteers might run well-known dedicated servers, like the public Tor relays [16]; alternatively, some or all client nodes could also play the server role in peer-to-peer deployment scenarios.

Ideally, clients and users would not need to trust any of the servers, as in the trust model SUNDR [27] implements for storage. D3 falls "just short" of this ideal: clients must trust that *at least one server* behaves honestly, but *need not know or choose which server(s) to trust.* We call this model *anytrust*, a term inspired by anycast communication, in which a sender sends a message to "any" of several recipients without knowing or caring about the actual destination.

Building on Dissent [14], which offered strong anonymity resistant to traffic analysis and disruption but limited scalability, D3 combines verifiable shuffles [28, 20, 9] with DC-nets techniques [11, 31, 14]. D3 uses a verifiable shuffle to set up a "schedule" for subsequent, more efficient DC-nets messaging rounds. D3 applies and benefits from the principles in anytrust for both its shuffle and messaging rounds.

In the Dissent shuffle, *every* group member must serially receive, randomly permute, decrypt, and resend a list of ciphertexts submitted by *all* members. In D3, in contrast, only the smaller set of servers participate in this inherently serial phase, reducing the latency and total communication complexity of the protocol when the number of servers is small relative to the number of users. The subsequent DC-nets messaging rounds also benefit from this trust model: in Dissent every group member shares a secret with all $N-1$ other members, and must compute and XOR together $N-1$ pseudo-random ciphertexts while transmitting. In D3, in contrast, the anytrust model requires clients share secrets only with the M servers, not with each other. Clients compute only $M \ll N$ ciphertexts during DC-nets messaging. Servers compute N cipherstreams (one per client) in messaging rounds, but this higher computation burden is more tolerable for well-provisioned servers. Finally, the ciphertext produced by clients during messaging does not depend

5th European Workshop on System Security, April 10, 2012, Bern, Switzerland.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
10 APR 2012

2. REPORT TYPE

3. DATES COVERED
00-00-2012 to 00-00-2012

4. TITLE AND SUBTITLE
Scalable Anonymous Group Communication in the Anytrust Model

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
**Naval Research Laboratory,4555 Overlook Ave.,
SW,Washington,DC,20375**

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
5th European Workshop on System Security, April 10, 2012, Bern, Switzerland.

14. ABSTRACT

Anonymous communication capabilities are useful and desirable but practical onion routing approaches are vulnerable to traffic analysis and DoS attacks?especially against a powerful adversary, such as a repressive government or compromised ISP. To fill this gap we introduce D3, the first practical anonymous group communication system offering anonymity and disruption resistance against strong traffic analysis and collusion attacks, with scalability to hundreds of group members and quick response to churn. D3 builds on a trust model we call anytrust. Anytrust is a decentralized client/server network model, in which each of many clients?representing group members?trust only that at least one of a smaller but diverse set of ?servers? or ?super-peers? behaves honestly, but clients need not know which server to trust. By combining and adapting verifiable shuffle and DC-nets techniques to anytrust, D3 achieves short shuffle latencies and efficient tree-based DC-nets ciphertext combining while guaranteeing message anonymity and integrity, transmission proportionality among group members, and adaptability to both network/node failures and active disruption. Experiments with a working prototype demonstrate that D3 is practical at scales infeasible in prior systems offering comparable security.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

on which other clients are online, while the smaller set of servers can adapt quickly in the event that a client becomes unresponsive (either unintentionally or maliciously): the servers need only to adjust their own ciphertext accordingly, enabling communication to proceed without interruption.

We have implemented a working prototype of D3 and validated it on Emulab and Deterlab, in networks of up to 60 real nodes and 576 virtual nodes. In a network of 56 nodes the D3 shuffle runs 10× faster than the original Dissent shuffle, and ongoing D3 communication rounds take 32× less CPU time than Dissent communication rounds of equal size. These results are significant improvements over the scalability of existing systems offering strong, traffic analysis resistant anonymity.

Section 2 motivates D3 and describes potential applications. Section 3 outlines the anytrust model and D3's design. Section 4 describes and evaluates the in-progress prototype implementation, and finally Section 6 concludes.

2. BACKGROUND AND MOTIVATION

This section motivates D3 by summarizing the *shuffled multicast* communication model it implements, and then discussing application scenarios in which this model may be useful.

2.1 Group Communication Model

As in Dissent [14], D3 offers anonymity *within a group*. The identities of a group's members may be well-known—e.g., the board members of an organization or club—and it may be known that a message came from *some member* of the group, but D3 prevents a message from being linked with the *particular* member who sent it, unless that link is deducible from the message's content. Complementary techniques such as MCONs [36] could help hide membership from outside observers, if desired, and Tor bridges [32] or decoy routing [25] could hide the use of anonymous communication technologies from detection or blocking in the network. These goals are out of this paper's scope, however.

D3 implements a *shuffled multicast* abstraction: communication proceeds in rounds according to an established schedule, and each group member presently online may send one message per round, which all other members receive in a random, shuffled order unknown to any participant. Unlike verifiable shuffles [28, 20], which assume messages are of some small, fixed size, D3 efficiently permits each member's message in a round to be empty or arbitrarily large. D3 ensures both message integrity—each group member receives every other member's unmodified message in each round—and proportionality—each member can submit one and only one message per round, and thus cannot tamper with group votes [24] or engage in sock-puppetry [34] without first obtaining multiple group memberships.

While D3 adopts a group communication model similar to Dissent, D3's *anytrust* client/server model enables D3 to support larger groups and tolerate network churn. These enhancements increase the "strength in numbers" that this group communication model can effectively offer, and potentially enable more diverse applications for which prior, less scalable anonymous group communication schemes would have been impractical. For example, Dissent [14] evaluated only large file transfers because the communication overhead in transferring small amounts of data in the DC-nets messaging phase was high and not a significant improvement over the shuffle alone. Herbivore [31] makes low latency guarantees (100s of seconds), but only for for small groups—generally less than 10 members.

2.2 Possible Applications for D3

Blogs and microblogs, such as Twitter[1], were widely used as organizational tools during the recent events in the Middle East, knowns as Arab Spring or Awakenings[2]. The risks inherent in these centralized services, however, have motivated attempts at building stronger anonymous communication tools, such as Tahrir[3]. Even these decentralized tools are generally susceptible to traffic analysis, however—an important concern when a country's ISPs are all effectively controlled by the government.

While D3 would be only one piece of a solution to this challenge, it could offer greater "strength in numbers" to microbloggers and ad-hoc political organizations. While an outspoken individual or small group may risk harsh punishment, such as jail-time or worse, a larger contingent—whose leaders can hide among their followers—might risk only Internet blocking or other "slap-on-the-wrist" punishments for using anonymity or circumvention software. D3 can protect vocal members who post sensitive or controversial material from being singled out from a larger group.

Reporters often use anonymous sources, but doing so can place the reporter under scrutiny and pressure. Using a system such as D3, a federation of reporters or a news organization could offer an anonymous information "hotline" enabling whistleblowers to connect with journalists, while offering both parties greater protection from subsequent pressure.

3. D3 SYSTEM DESIGN

This section outlines D3's architecture and design at a high level, first covering the anytrust model, then sketching group definition, session management, session setup via verifiable shuffles, and messaging rounds via DC-nets.

3.1 The Anytrust Model

In existing, practical group anonymity protocols [31, 14], all members typically interact directly as equals. These systems work at small scales, but as the group becomes larger and more diverse, these systems become impractical and unsustainable. We introduce *anytrust* as a means to scale standard anonymity protocols relying on group communication. Anytrust shifts the heaviest computation and communication overheads to a smaller but diverse subset of well-provisioned members, who are "trusted" only minimally and collectively.

In the anytrust model, a *client* represents a user interested in consuming a service. Nodes that facilitate the service we call *servers*. We use the terms "client" and "server" loosely, independent of particular applications or deployment scenarios. For example, a group's servers might be well-known, dedicated machines run by volunteers or organizations wishing to support anonymous communication online, analogous to the public Tor relays [16]. Alternatively, the "server" nodes might in fact also be "client" nodes doing double-duty, so that the servers are merely a subset of the clients. In a purely "peer-to-peer" deployment model, the servers might represent "super-peers" chosen in some automated fashion from the set of clients.

How the servers are chosen can obviously affect group security, and automated selection mechanisms present potential attack vulnerabilities that we do not address here. For now, we merely assume that the set of servers is "given" and pursue maximum security in that framework.

[1] http://www.twitter.com

[2] http://arabawakenings.thestar.com/

[3] https://github.com/sanity/tahrir/wiki/Overview

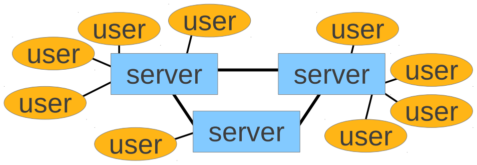

Figure 1: D3's client/server model.

The most security-critical assumption anytrust makes about the servers is that *at least one server is honest*: i.e., that at least one server runs the anonymity protocol as specified, and does not save, leak, or share information it is supposed to hide or destroy. Although we assume that at least one server is honest, *we do not assume that clients "know"—or can "guess"—which servers are honest.* Even if a client communicates with the group via a dishonest server, that dishonest server cannot compromise the client's anonymity, even if that server colludes with all but one of the other servers. This anytrust model contrasts with many of the security assumptions made by existing practical anonymity systems such as Tor [16] or Herbivore [31], where a small number of "wrong" choices—e.g., the choice of entry and exit relay in Tor—can completely compromise a client's anonymity. While anytrust relies for scalability on the number of servers being "small," a set of, say, 10 diverse servers chosen appropriately can offer reasonable performance under anytrust, and may offer much greater security than a client's typical choice of 3 relay nodes in Tor for example (only two of which need to be bad to compromise the client).

3.2 D3 Overlay Network Model

D3 assumes that clients are unreliable and that they have limited network connectivity and computational power. We assume that clients can connect to servers, but not necessarily to other clients.

D3 assumes that servers, in contrast, are well-provisioned, reliable, and connected to all other servers. Techniques such as resilient overlay networks [3] can help ensure server connectivity.

Finally, D3 leverages this client/server asymmetry to respond quickly to client churn. When clients come and go, D3 assumes that the servers can interact with each other immediately to adjust, but we wish to avoid requiring the servers to interact with the clients in order to do so, as would be necessary in a conventional "flat" anonymous group communication system. Figure 1 demonstrates an overlay organized according to this model.

3.3 Group Definition and Sessions

A D3 group consists of a set of well-known identities represented by public keys. D3 assumes that any group has an agreed-upon *definition*: e.g., a file listing the public keys and human-readable names of the clients (group members), the public keys and IP addresses or hostnames of the servers supporting the group, and any relevant group management policy settings. A group's policy might specify a *quorum*, for example: the minimum number of clients that must be online in order for the servers to allow communication to proceed, guaranteeing the clients a minimum anonymity set size. We assume all clients and servers agree on a group's definition: the easiest way to accomplish this is for the "group identifier" to be a cryptographic hash of the file defining the group, making group identifiers self-certifying [19], and guaranteeing that all nodes who "think" they're talking about the same group also agree on the group's membership and policy.

The fundamental unit of anonymous communication is a *session*, wherein a dynamic group participates in a series of anonymous exchanges until the group size becomes too small to offer sufficient anonymity (e.g., below the group's quorum). Within a session, the anonymous exchanges occur inside rounds. Depending on the construction of the anonymity protocol, either a single or many anonymous exchanges may take place in a given round.

In the current design, a session is initiated by a distinguished group member called the *leader*. The leader coordinates the session, handling member registration, running the shuffle-based setup process below, initiating and driving subsequent DC-nets messaging rounds, and confirming the expulsion or departure of members throughout the process. The leader is currently trusted for group availability, but *not* for anonymity or other security properties. If the leader is faulty or malicious, the group may become unusable for communication, but no safety properties are compromised. Standard leader election techniques could remove this obvious DoS vulnerability, but we leave this refinement to future work.

Once the leader initiates a session, other online peers (both clients and servers) join, and prove themselves legitimate and unique members of the group, by proving ownership of one of the public keys listed in the group definition. Once a sufficient number of clients and servers have joined according to group policy, the leader initiates the shuffle protocol below to commence group communication.

3.4 Session Setup via Verifiable Shuffle

As in Dissent [14], D3 uses a verifiable shuffle as a "setup phase" for subsequent DC-nets communication. In Dissent, each group member must first choose a message to send, form a *descriptor* containing the length and cryptographic information about its message, wrap its message in $2N$ layers of onion encryption—two layers for each group member—and submit the encrypted messages to a verifiable shuffle protocol. Since these shuffled descriptors are cryptographically bound to the messages to be sent, this design limits the group to one DC-nets round per shuffle.

In D3, all group members first create fresh public/private keypairs, called *pseudonym keys*, then the group shuffles this set of public pseudonym keys instead of message-specific descriptors. D3 uses this shuffle to create a random permutation unknown to everyone and to assign each member a verifiable "slot"—the position of its pseudonym key in the secret permutation. After the shuffle, each member knows its own position, by recognizing its own public pseudonym key, but no member knows which pseudonym keys correspond to other (honest, non-colluding) members.

By shuffling keys instead of message descriptors as in Dissent, D3 can "re-use" the same shuffle for multiple subsequent messaging rounds, which is important because the shuffle is both more expensive and has higher latency than messaging rounds.

To ensure that the resulting shuffle is a truly *random* permutation unknown to any one member, in the presence of dishonest nodes, multiple nodes must participate in the shuffle "redundantly." In the Dissent shuffle, *every* group member shuffles *every* message descriptor, analogous to a mix-network in which every message always passes through every relay. These shuffles are inherently serial: one shuffler must shuffle a whole "batch" of cyphertexts and decrypt one "onion layer" before passing the whole batch onto the next shuffler. This serial process incurs high latency as group size increases, limiting Dissent's scalability. D3 leverages anytrust to address this limitation: all N clients submit ciphertexts to the shuffle (namely their onion-encrypted pseudonym keys), but *only* the $M \ll N$ servers actually shuffle them, resulting in a fundamental latency of $O(M)$ rather than $O(N)$. Further, each client need only "wrap" its pseudonym key in M layers of encryption rather than N before the shuffle. By the anytrust assumption that *some* server is honest and not colluding, the single honest server's participation in the shuffle ensures that the entire batch of ciphertexts is randomly permuted, *even if all other servers are compromised and the honest clients do not know which server is honest.*

D3 uses an anytrust derivation of the online-verifiable shuffle introduced by Brickell and Shmatikov [9], which is attractive due to its simplicity and by making use of only "off-the-shelf" cryptographic algorithms such as RSA-OAEP. D3 could in principle use any verifiable shuffle, however, such as one of the cryptographic shuffle schemes permitting offline verification [28, 20]. We refer to this prior work for details on implementation of verifiable shuffles.

3.5 DC-nets Messaging Rounds

Once a session is set up via the verifiable shuffle, the leader commences *messaging rounds* on a schedule defined by group policy. For example, policy might instruct the leader to initiate the next messaging round immediately after the previous one has finished, to minimize latency for interactive applications, or to initiate rounds at occasional intervals to conserve network bandwidth for more delay-tolerant applications. To resist traffic analysis, however, scheduling and initiation of rounds must be done independently of any given member's "desire" to send a message: otherwise a powerful adversary could trivially tell which member(s) "wanted" to send in a given messaging round based on which member(s) performed a network-level session initiation.

As in Dissent, D3 uses DC-nets communication [11] for efficient communication during messaging rounds, on a schedule defined by the shuffle. In existing DC-nets protocols [14, 31], every group member shares a secret with all $N-1$ other members, computes $N-1$ pseudorandom strings seeded with these shared secrets, then XORs these strings with each other and with any cleartext to be transmitted. A group leader then collects and XORs all members' ciphertexts; the pseudorandom strings cancel out because each is included exactly twice, leaving only the cleartext.

Classic DC-nets suffer the challenges of anonymous jamming and scalability. Dissent addressed the jamming problem using the schedule set up by the verifiable shuffle, an idea D3 retains and extends. Some modification is required to allow multiple messaging rounds to follow a single shuffle, but we omit these details for the sake of brevity.

To address the scalability limitations of classic DC-nets, D3 again leverages the anytrust model. In place of "all-to-all" secret sharing, D3 clients share secrets only with servers and vice versa. This change preserves DC-nets' computational anonymity in the anytrust model, because if there is at least one honest server, each honest client shares a secret with that honest server (even though the client doesn't know which server is honest). Through their keys shared with the honest server, transmissions of every *pair* of honest clients in turn become indistinguishable from random bits by any other colluding set of clients and servers.

While D3 thus offers the same anonymity guarantees as an implementation of classic DC-nets under the anytrust assumption, D3 clients need to compute only $M \ll N$ pseudorandom strings. Servers must compute N strings, one per client, but we assume that servers will be able to handle this higher computational load. Just as importantly for robustness, since clients do not share secrets with each other, a client's ciphertext *does not depend on the set of other clients currently online*. This means that when a client goes offline, the servers need only cooperate with each other to adjust their ciphertext computations accordingly, and need not restart the ciphertext collection process iteratively, which would lead to many DoS vulnerabilities from unreliable or malicous clients.

If a client or server computes and submits an incorrect ciphertext, e.g., in an attempt to disrupt the DC-net channel, D3 implements an *accusation process*, which enables the group to identify and expel the culprit. After k DC-net messaging rounds, all honest group members identify a faulty client or server with probability $\geq 1-(\frac{1}{2})^k$. We omit the details of the accusation process but we emphasize that the DC-net protocol, like the shuffle protocol, maintains accountabilty.

3.6 Design Limitations

While we believe D3 is a step toward building secure, practical, and scalable anonymity systems, it still has many limitations.

D3 assumes that all servers know and can verify the well-known identities of all clients as they join and leave a session, and thus know which clients are online in each round. If an adversary has compromised at least one server, and clients use D3 to send messages that are linkable over time (e.g., by signing them with a long-lived "pseudonym key"), the adversary can use intersection attacks [37] to narrow the set of identities a given pseudonym might correspond to. There are ways D3 can be used to minimize such dangers: e.g., in voting applications, where messages are simple "yes/no" ballots, these messages are not likely to be linkable.

Another choice is to configure a group's quorum so that *all* clients must be present in order for communication to proceed. Still another approach is not to allow clients to rejoin once they have ever gone offline, and cease group communication if the online set goes below the quorum. This solution might be suitable for short-lived groups such as anonymous online auctions for example. Nevertheless, intersection attacks represent a significant ongoing challenge, which we leave for future work.

Other challenges are more obvious: for example, while we demonstrate below that D3 can scale to hundreds of nodes, true "strength in numbers" in many situations may require anonymity sets of thousands or more; scaling D3 to these levels will likely require additional work.

4. IMPLEMENTATION AND EVALUATION

We have implemented D3 using C++, the Qt framework, and CryptoPP. The prototype supports the original Dissent shuffle and DC-net (bulk) [14] as well as the D3 shuffle and DC-net. The implementation also incorporates session management, a minimal PeerReview component [23], and a simple peer-to-peer overlay that ensures full connectivity in the presence of network asymmetries.

Our prototype does not handle long-term public key management, nor does it gracefully adapt to non-responsive nodes. We have implemented the full "blame" process for the original Dissent protocol and for our client / server shuffle and DC-net protocols for a "flat" (broadcast) network topology. We have not yet implemented the blame protocol for the XOR tree topology.

We have evaluated our prototype on DeterLab [15] and Emulab [18]. Multiple virtual D3 nodes in separate processes run on each physical testbed node. Each testbed node connects to a central switch, via virtual links with 10Mbps bandwidth and 50ms delay.

4.1 Scalable Shuffle

We compared Dissent's shuffle [14] to D3's scalable shuffle, in Figure 2, using various network sizes with a fixed message size of 8 KB. For scalable shuffle runs, the server node count was fixed at 10 or the set of all nodes, whichever was smaller. As the network size increased, the scalable shuffle performed significantly better than the original shuffle. For example, at a network size of 64 nodes, the scalable shuffle completed 10× faster than the original shuffle did.

In the original shuffle, each additional participant requires an additional non-parallelizable communication and decryption round. As network size increases, the cost of these rounds dominates overall protocol runtime. With a 64-node network, these rounds con-

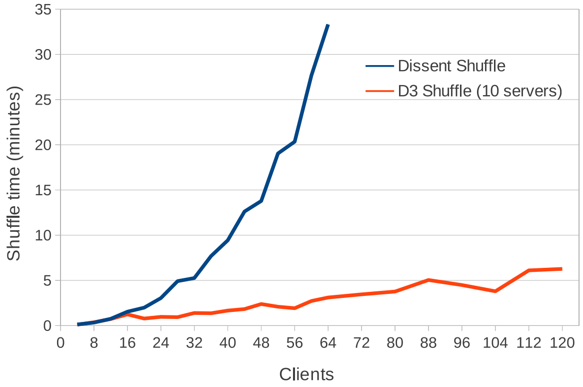

Figure 2: Comparison of the original Dissent shuffle to the client / server shuffle (with 10 servers) with nodes exchanging 8 KB messages.

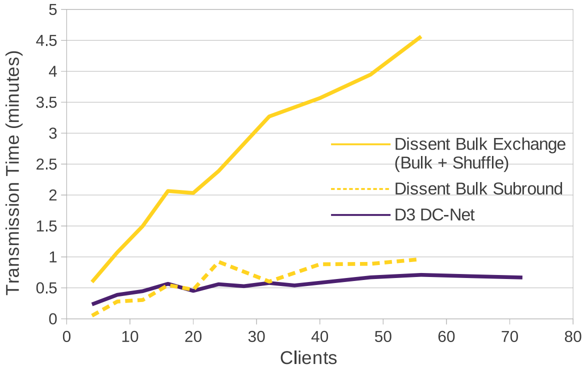

Figure 3: Comparison of the original Dissent bulk protocol to the client / server DC-net protocol, with one node transmitting a 256 KB message.

sumed over 90% of the total protocol runtime. The scalable shuffle fixes the non-parallelizable communication rounds to a constant equal to the number of server nodes, while adding user nodes increases the total communication cost (quadratic in the number of users) and total computation cost (linear in the number of users).

4.2 Scalable DC-Net

To compare D3's scalable DC-net to the Dissent bulk protocol, we simulated a series of exchanges wherein a single node continuously sent a 256 KB message, while all other nodes send nothing. This traffic pattern approximates the traffic that might be generated from an anonymous blog or media broadcast: e.g., a citizen journalist publishing a series of JPEG images from an ongoing protest. Because the test focuses on comparing DC-net exchanges, both use the client / server shuffle.

The results in Figure 3 validate D3's key design decisions: the anytrust model, tree-based multicast, and repeatable DC-net exchanges. Dissent's efficiency could be increased by bundling more images together per round, but this would result in longer delays and may incur retries due to intermediate disconnections. D3's shorter inter-exchange delay reduces the impact of disconnections.

For a network of 576 virtual D3 nodes, as Figure 4 indicates, a D3 message transmission completes in just over two minutes. These tests were performed by running multiple D3 nodes on single-core Pentium III machines (and a few dual-core Xeon machines) with inter-machine bandwidth of 1Mbps. The limit of 576 nodes was goverened solely by the number of testbed machines available rather than design factors inherent to the protocol. With more machines, we expect that the D3 network size could scale further.

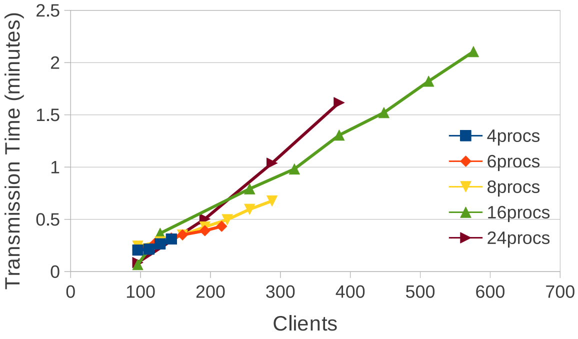

Figure 4: Transmission time for a D3 DC-net protocol run in which one user sends a 256-byte message, varying the network size and number of virtual D3 nodes per machine.

5. RELATED WORK

The major design tradeoff in anonymity protocols is between performance and anonymity. The anonymity protection of existing protocols spans the full range from very strong to very weak, with speed and reliability improving as the anonymity decreases.

DC-nets [11] implement anonymous group broadcast and offer strong anonymity guarantees. Unfortunately, their high communication overhead and their vulnerability to denial-of-service attacks makes them largely impractical. Herbivore [31] attempts to make DC-nets more scalable, but it provides unconditional anonymity only among small groups of nodes. Other approaches [22] provide some DoS protection but do not protect against collisions.

Mix networks [10] generally provide slightly weaker anonymity but have lower communication costs. Users submit encrypted messages to a network of mixes, each of which stores the messages, modifies their encryption to prevent tracking, mixes them, and then forwards them through the network. Users can be vulnerable to traffic analysis and several active attacks [30, 17].

Onion routing [21] is a popular scheme that provides anonymity against relatively weak adversaries, with latency and communication costs comparable to non-anonymous communication. In onion routing, users *onion encrypt*, i.e. multiply encrypt, their messages, and send them through a short path of onion routers, each of which removes a layer of encryption. Users are vulnerable to traffic analysis and choosing malicious routers [35], but the relatively high performance has made this approach attractive and popular in many situations (e.g., Tor [16], I2P [1], GNUnet [6])

The Dissent shuffle [14], upon which D3 is based, provides denial-of-service protection and strong anonymity, but is inefficient for long-term communication within a group of nodes, and communication overhead is prohibitive when used among large groups.

6. CONCLUSION

In this paper, we presented the anytrust model as a tool to enable strong, scalable group anonymity systems. We demonstrated the model's usefulness via D3, which incorporates the anytrust principle into shuffle and DC-net anonymity protocols. The D3 shuffle significantly improves upon the original Dissent shuffle by replacing the N serial communication rounds with a constant number of rounds. By applying an anytrust tree model to a DC-net exchange, we see that performance boosts become apparent in network sizes as small as 40. For future work, we will explore ways in which we can further reduce the D3 DC-net message costs and latencies, and to protect against higher-level vulnerabilities such as long-term intersection attacks.

Acknowledgments

We wish to thank Vitaly Shmatikov, Paul Syverson, and the anonymous reviewers for helpful discussion. This material is based upon work supported by the Defense Advanced Research Agency (DARPA) and SPAWAR Systems Center Pacific, Contract No. N66001-11-C-4018. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Agency (DARPA) and SPAWAR Systems Center Pacific.

7. REFERENCES

[1] Invisible internet project (i2p), Jan. 2012. http://www.i2p2.de/.

[2] B. Adida. *Advances in cryptographic voting systems*. PhD thesis, Cambridge, MA, USA, 2006. Adviser-Rivest, Ronald L.

[3] D. G. Andersen et al. Resilient overlay networks. In *18th SOSP*, Oct. 2001.

[4] J. M. Balkin. Digital speech and democratic culture: A theory of freedom of expression for the information society. *Faculty Scholarship Series*, 2004. Paper 240.

[5] K. Bennett and C. Grothoff. GAP — practical anonymous networking. In *PET*, Mar. 2003.

[6] K. Bennett and C. Grothoff. gap - practical anonymous networking. In *Designing Privacy Enhancing Technologies*, page 141–160, 2003.

[7] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *18th International World Wide Web Conference*, pages 551–551, April 2009.

[8] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz. Denial of service or denial of security? How attacks on reliability can compromise anonymity. In *14th ACM CCS*, Oct. 2007.

[9] J. Brickell and V. Shmatikov. Efficient anonymity-preserving data collection. In *12th KDD*, Aug. 2006.

[10] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), Feb. 1981.

[11] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, Jan. 1988.

[12] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Workshop on Design Issues in Anonymity and Unobservability*, July 2000.

[13] A. Comninos. Twitter revolutions and cyber crackdowns: User-generated content and social networking in the Arab spring and beyond, June 2011.

[14] H. Corrigan-Gibbs and B. Ford. Dissent: accountable anonymous group messaging. In *CCS*, pages 340–350, Oct. 2010.

[15] Deterlab cybersecurity testbed. http://isi.deterlab.net/.

[16] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *13th USENIX Security Symposium*, Berkeley, CA, USA, 2004.

[17] R. Dingledine and P. Syverson. Reliable MIX cascade networks through reputation. In *Financial Cryptography*, Mar. 2002.

[18] Emulab network emulation testbed. http://emulab.net/.

[19] K. Fu, M. F. Kaashoek, and D. Mazières. Fast and secure distributed read-only file system. *TOCS*, 20(1):1–24, Feb. 2002.

[20] J. Furukawa and K. Sako. An efficient scheme for proving a shuffle. In *CRYPTO*, Aug. 2001.

[21] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding Routing Information. In *Information Hiding Workshop*, May 1996.

[22] P. Golle and A. Juels. Dining cryptographers revisited. *Eurocrypt*, May 2004.

[23] A. Haeberlen, P. Kouznetsov, and P. Druschel. PeerReview: Practical accountability for distributed systems. In *21st SOSP*, Oct. 2007.

[24] H. Hsieh. Doonesbury online poll hacked in favor of MIT. *MIT Tech*, 126(27), June 2006.

[25] J. Karlin et al. Decoy routing: Toward unblockable internet communication. In *FOCI*, Aug. 2011.

[26] S. F. Kreimer. Technologies of protest: Insurgent social movements and the first amendment in the era of the internet. *University of Pennsylvania Law Review*, 150:119–171.

[27] J. Li, M. Krohn, D. Mazières, and D. Shasha. Secure untrusted data repository (SUNDR). In *6th OSDI*, Dec. 2004.

[28] C. A. Neff. A verifiable secret shuffle and its application to e-voting. In *8th CCS*, pages 116–125, Nov. 2001.

[29] J.-F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In *Design Issues in Anonymity and Unobservability*, July 2000.

[30] A. Serjantov, R. Dingledine, and P. Syverson. From a trickle to a flood: Active attacks on several mix types. *Information Hiding*, pages 36–52, 2003.

[31] E. G. Sirer et al. Eluding carnivores: File sharing with strong anonymity. In *11th SIGOPS European Workshop*, Sept. 2004.

[32] R. Smits et al. BridgeSPA: Improving tor bridges with single packet authorization. In *WPES*, Oct. 2011.

[33] E. Stein. Queers anonymous: Lesbians, gay men, free speech, and cyberspace. *Harvard Civil Rights-Civil Liberties Law Review*, 38(1), 2003.

[34] B. Stone and M. Richtel. The hand that controls the sock puppet could get slapped. *New York Times*, July 2007.

[35] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr. Towards an Analysis of Onion Routing Security. In *Design Issues in Anonymity and Unobservability*, July 2000.

[36] E. Vasserman, R. Jansen, J. Tyra, N. Hopper, and Y. Kim. Membership-concealing overlay networks. In *16th ACM CCS*, Nov. 2009.

[37] M. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *Proceedings of NDSS '02*, February 2002.